Kryptografia

Laboratorium - lista nr 3, 30 III

Zadanie 1 (1 pkt) Napisz program, który generuje klucz AES i zapisuje go do keystore'a (skorzystaj np. z biblioteki Bouncy Castle).

Zadanie 2 (2 pkt) Napisz program, który szyfruje/deszyfruje algorytmem AES wskazany plik/pliki na dysku. Parametrami programu są:

- tryb szyfrowania OFB/CTR/... (w szczególności ma wspierać AES_cbc_encrypt, możesz skorzystać z *openssl*),
- ścieżka do keystore'a przechowującego klucz,
- identyfikator klucza.

Hasło do klucza z keystore'a należy wczytać interaktywnie.

Program ma pracować w dwóch trybach:

encryption oracle (wyrocznia szyfrująca) – na wejściu wczytuje zbiór wiadomości m^i i zwraca odpowiadające im kryoptogramy c^i.

challenge – bierze na wejściu dwie wiadomości m_0, m_1, generuje losowy bit b i zwraca odpowiadający kryptogram c_b.

Zadanie 3 (7 pkt) Zaimplementuj adwersarza (CPA-distinguisher), który rozróżnia z niepomijalnym prawdopodobieństwem zmodyfikowaną wersję AES_cbc_encrypt dla ataku chosen-plaintext. Załóż, że program zaimplementowany w zadaniu 2 generuje IV poprzez zwiększanie jej wartości o 1 przy każdym szyfrowaniu wiadomości (vide: Lista 3 zadanie 4).

Możesz wymusić takie działanie poprzez dostarczenie odpowiedniej wartości *ivec* dla (*include/openssl/aes.h*):

void *AES_cbc_encrypt*(const unsigned char ∗*in*, unsigned char ∗*out*,
size_t *length*, const AES_KEY ∗*key*,
unsigned char ∗*ivec*, const int *enc*);